

**Amendments to the Claims**

Please cancel claims 11-20. The currently pending claims after amendment are listed below.

1        1.        (Previously Presented) A method for analyzing a computer program, comprising the steps  
2        of:  
3                defining a plurality of breakpoints for said computer program, at least one of said  
4        breakpoints including a respective expected code path condition;  
5                executing said computer program;  
6                with respect to each breakpoint including a respective expected code path condition  
7        encountered during execution of said computer program, automatically determining whether an  
8        actual code path taken during execution of said computer program matches the respective  
9        expected code path condition of the encountered breakpoint; and  
10               automatically halting execution of said program if said actual code path taken during  
11       execution of said computer program does not match the expected code path condition  
12       regardless of the condition of any key variable specified as a condition of the conditional  
13       breakpoint.

1        2.        (Original) The method for analyzing a computer program of claim 1, wherein said step of  
2        defining a plurality of breakpoints comprises, for said at least one breakpoint including a  
3        respective expected code path condition, defining a respective set of expected predecessor  
4        breakpoints.

1        3.        (Original) The method for analyzing a computer program of claim 1, wherein said step of  
2        defining a plurality of breakpoints comprises, for said at least one breakpoint including a  
3        respective expected code path condition, defining a respective set of basic blocks required to be  
4        encountered during execution before the respective breakpoint.

1 4. (Original) The method for analyzing a computer program of claim 1, wherein said step of  
2 defining a plurality of breakpoints comprises, for said at least one breakpoint including a  
3 respective expected code path condition, defining a respective set of basic blocks prohibited from  
4 being encountered during execution before the respective breakpoint.

1 5. (Original) The method for analyzing a computer program of claim 1, wherein said step of  
2 defining a plurality of breakpoints comprises, for at least one of said breakpoints including a  
3 respective expected code path condition, at least one respective additional condition.

1 6. (Original) The method for analyzing a computer program of claim 5, wherein said step of  
2 halting execution of said program is performed if said actual code path taken during execution of  
3 said computer program does not match the expected code path condition, or if said at least one  
4 respective additional condition is met.

1 7. (Original) The method for analyzing a computer program of claim 1, wherein said step of  
2 defining a plurality of breakpoints comprises the steps of:  
3 displaying code statements of said computer program to a user;  
4 interactively receiving a user selection of a code statement as the location of a breakpoint  
5 having an expected code path condition;  
6 responsive to receiving said user selection, displaying code statements which necessarily  
7 execute before the selected code statement in a highlighted form; and  
8 interactively receiving a user selection of said expected code path condition.

1       8.     (Original) The method for analyzing a computer program of claim 7, wherein said step of  
2     interactively receiving a user selection of said expected code path condition comprises:  
3         interactively receiving a user selection of a code statement as the location of a basic block  
4     of code with which an expectation with respect to code path during execution is associated;  
5         responsive to receiving said user selection of a code statement as the location of a basic  
6     block, determining the basic block to which the selected code statement belongs;  
7         displaying code statements in the basic block containing said selected code statement in a  
8     highlighted form.

1       9.     (Original) The method for analyzing a computer program of claim 8, further comprising:  
2         interactively receiving a user indication of said expectation with respect to code path  
3     during execution associated with the basic block containing the selected code statement;  
4         wherein said step of displaying code statements in the basic block containing the selected  
5     code statement in a highlighted form comprises displaying said statements in a first highlighted  
6     form responsive to a user indication of a first expectation with respect to code path during  
7     execution, and displaying said statements in a second highlighted form responsive to a user  
8     indication of a second expectation with respect to code path during execution, said second  
9     highlighted form being visually distinct from said first highlighted form.

1        10.    (Original) The method for analyzing a computer program of claim 1,  
2                wherein said step of defining a plurality of breakpoints comprises generating a respective  
3        breakpoint definition record for each of said plurality of breakpoints, said breakpoint definition  
4        record being separate from executable code of said computer program; and  
5                wherein said step of executing said computer program includes monitoring said program  
6        during execution to determine whether a breakpoint has been encountered, and invoking a  
7        breakpoint handler responsive to encountering a breakpoint during execution, said breakpoint  
8        handler performing said step of determining whether an actual code path taken during execution  
9        of said computer program matches the respective expected code path condition of the encountered  
10       breakpoint.

11 - 20.    (Cancelled)